IN THE CLAIMS

1. (Currently amended) A processing unit, comprising:

a plurality of processor clusters, each processor cluster having an output to send a signal representing instruction requests, an instruction request responsive to a cache miss by a processing unit within the processor clusters;

an instruction request arbiter, having an input coupled to the outputs of the plurality of processor clusters, the instruction request arbiter controlling access of the plurality of processor clusters to submit instruction requests by broadcasting a [[the]] traffic mode to the plurality of processor clusters, the instruction request arbiter having a second output coupled to the inputs of the pluratlity of processor clusters and controlling access to instruction request submissions from the plurality of processor clusters by broadcasting a signal representing a high traffic mode responsive to an average number of instruction requests exceeding a limit, wherein during a high traffic mode, the plurality of processor clusters submit instruction requests during scheduled slots, the instruction request arbiter also detecting conflicts between the instruction requests; and

an instruction memory, having an input coupled to a first output of the instruction request arbiter, the instruction memory sending a signal representing instruction data to the plurality of processor clusters responsive to receiving non-conflicting instruction requests from the instruction request arbiter.

2. (Original) The processing unit of claim 1, wherein the instruction request arbiter resolves conflicts between instruction requests having different priority indications by selecting an instruction request with the highest priority.

3. (Original) The processing unit of claim 2, wherein a high priority is indicative of an instruction associated with a critical instruction request.

4. (Original) The processing unit of claim 2, wherein the instruction request arbiter increments the priority indication of an unselected instruction request.

5. (Original) The processing unit of claim 1, wherein the instruction request arbiter resolves conflicts between instruction requests having equal priority indications by using round-robin arbitration.

6. (Original) The processing unit of claim 5, wherein the instruction request arbiter increments the priority indication of an unselected instruction request.

7. (Original) The processing unit of claim 1, wherein the instruction request arbiter detects the conflicting instruction requests when each instruction request is associated with the same group of cache sets of the instruction memory.

8. (Canceled)

9. (Original) The processing unit of claim 1, wherein the instruction request arbiter, having a second output coupled to inputs of the plurality of processor clusters, the instruction request arbiter controlling access to instruction request submissions by broadcasting a signal representing a low traffic mode responsive to an average number of instruction requests within a limit, wherein during low traffic mode, the processor clusters submit instruction requests during empty slots.

10. (Original) The processing unit of claim 1, further comprising an instruction request bus, coupled to a second instruction request arbiter output, coupled the plurality of processor cluster outputs, and coupled to the instruction request arbiter input, the instruction request bus comprising a plurality of flip flops to pipeline the instruction requests and the traffic mode.

11. (Original) The processing unit of claim 10, wherein the instruction request bus comprises a slotted ring.

12. (Original) The processing unit of claim 1, wherein the processing unit comprises a network processor and the instruction data comprises packet processing instructions related to at least one from the group consisting of: packet routing, switching, bridging, and forwarding.

13. (Original) The processing unit of claim 1, wherein the plurality of processor clusters comprise a plurality of processing units, each of the processing units submitting requests responsive to a traffic mode.

14. (Original) The processing unit of claim 1, wherein at least one of the plurality of processor clusters comprises at least one multithreaded processing unit capable of processing a plurality of instruction threads, each of the plurality of threads submitting instruction requests responsive to a traffic mode.

15. (Original) The processing unit of claim 14, wherein the at least one multithreaded processing unit comprises a plurality of multithreaded processing cores, each of the plurality of multithreaded processing cores submitting instruction requests responsive to a traffic mode.

16. (Currently amended) A processing unit, comprising:

a plurality of means for cluster processing, each means for cluster processing sending instruction requests, an instruction request responsive to a cache miss by a means for processing within the means for cluster processing;

means for arbitration, coupled to the plurality of means for cluster processing, the means for arbitration controlling access of the plurality of means for cluster processing to submit instruction requests by broadcasting a [[the]] traffic mode to the plurality of means for cluster processing, the means for arbitration having a second output coupled to the inputs of the pluratlity of means for cluster processing and controlling access to instruction request submissions from the plurality of means for cluster processing by broadcasting a signal representing a low traffic mode responsive to an average number of instruction requests within a limit, wherein during a low traffic mode, the plurality of means for cluster processing submit instruction requests during empty slots, the means for arbitration also detecting conflicts between the instruction requests; and

means for storing, coupled to the means for arbitration, the means for storing sending instruction data to the plurality of means for cluster processing responsive to receiving non-conflicting instruction requests from the means for arbitration.

17. (Original) The processing unit of claim 16, wherein the means for arbitration resolves conflicts between instruction requests having different priority indications by selecting an instruction request with the highest priority.

18. (Original) The processing unit of claim 16, wherein the means for arbitration resolves conflicts between instruction requests having equal priority indications by using round-robin arbitration.

19. (Original) The processing unit of claim 16, wherein the means for arbitration detects the conflicting instruction requests when each instruction request is associated with the same group of cache sets of the means for storing.

20. (Original) The processing unit of claim 16, wherein the processing unit is a network processor and the instruction data comprises packet processing instructions related to at least one from the group consisting of: packet routing, switching, bridging, and forwarding.

21. (Currently amended) In a processing unit having an instruction memory hierarchy, a method for distributing instructions to an plurality of processing units organized in a plurality processor clusters, comprising:

determining a traffic mode for the plurality of processor clusters, an instruction request responsive to a cache miss by a processing unit within a processor cluster;

controlling submissions of instruction requests by each of the processing units by broadcasting the traffic mode to the plurality of processor clusters, including broadcasting a signal representing a high traffic mode responsive to an average number of instruction requests exceeding a limit, wherein during a high traffic mode, the plurality of processor clusters submit instruction requests during scheduled slots;

receiving instruction requests;

detecting conflicts between the instruction requests; and

sending instruction data responsive to non-conflicting instruction requests.

22. (Original) The method of claim 21, further comprising:

resolving conflicts between instruction requests having different priority indications by selecting an instruction request with the highest priority.

23. (Original) The method of claim 21, wherein a high priority is indicative of an instruction associated with a critical instruction request.

24. (Original) The method of claim 22, further comprising:
incrementing the priority indication of an unselected instruction request.

25. (Original) The method of claim 21, further comprising:
resolving conflicts between instruction requests having equal priority indications by using round-robin arbitration.

26. (Original) The method of claim 25, further comprising:
incrementing the priority indication of an unselected instruction request.

27. (Original) The method of claim 21, wherein the detecting the conflicting instruction requests comprises detecting instruction requests associated with a same group of cache sets of an instruction memory.

28. (Canceled)

29. (Original) The method of claim 21, wherein the receiving the instruction requests comprises receiving the instruction request during empty slots responsive to a low traffic mode, the low traffic mode responsive to an average number of instruction requests within a limit.

30. (Original) The method of claim 21, wherein the receiving the instruction requests comprises receiving a plurality of instruction requests from a plurality of threads associated with the plurality of processing units.

31. (Original) The method of claim 30, wherein the receiving the instruction requests comprises receiving a plurality of instruction request from a plurality of threads associated with a plurality of multithreaded processing cores within at least one of the plurality of processing units.

32. (Currently amended) An instruction request arbiter, comprising:
means for bus accessing, coupled to receive an indication of instruction requests, the means for bus accessing determining a traffic mode for a plurality of means for cluster processing, the means for bus accessing controlling submissions of

instruction requests by broadcasting the traffic mode to the plurality of means for cluster processing, the means for bus accessing having a second output coupled to the inputs of the pluratlity of means for cluster processing and controlling access to instruction request submissions from the plurality of means for cluster processing by broadcasting a signal representing a high traffic mode responsive to an average number of instruction requests exceeding a limit, wherein during a high traffic mode, the plurality of means for cluster processing submit instruction requests during scheduled slots, an instruction request responsive to a cache miss by a means for processing within the means for cluster processing; and

means for memory accessing, coupled to receive instruction requests, the means for memory accessing forwarding non-conflicting instruction requests.

33. (Original) The instruction request arbiter of claim 32, wherein the means for memory accessing resolves conflicts between instruction requests having different priority indications by selecting an instruction request with the highest priority.

34. (Original) The instruction request arbiter of claim 33, wherein the means for memory accessing increments the priority indication of an unselected instruction request.

35. (Original) The instruction request arbiter of claim 32, wherein the means for memory accessing resolves conflicts between instruction requests having equal priority indications by using round-robin arbitration.

36. (Original) The instruction request arbiter of claim 35, wherein the means for memory accessing increments the priority indication of an unselected instruction request.

37. (Original) The instruction request arbiter of claim 32, wherein the memory accessing means detects conflicting instruction requests where more than one instruction request is associated with a same group of cache sets of a storing means.

38. (Original) The instruction request arbiter of claim 32, wherein the high priority is indicative of an instruction associated with a critical instruction request.

39. (Canceled)

40. (Original) The instruction request arbiter of claim 32, wherein the means for bus accessing controls submissions of instruction requests by the plurality of means for cluster processing with a low traffic mode responsive to an average number of instruction request indications within a limit, the low traffic mode indicating that the plurality of means for cluster processing can submit instruction requests during an empty slot.

41. (Original) The instruction request arbiter of claim 32, wherein the plurality of means for cluster processing each comprise a plurality of means for processing, and the means for bus accessing controls submissions of instruction requests by each of the means for processing.

42. (Original) The instruction request arbiter of claim 41, wherein at least one of the plurality of means for processing comprises a plurality of means for threading, and the means for bus accessing controls submissions of instruction requests by each means threading.

43. (Currently amended) An instruction request arbiter, comprising:

a bus access module, having an input coupled to receive signals representing a plurality of instruction request indications, the bus access module determining a traffic mode to control submissions of instruction requests by processor clusters, the bus access module controlling submissions of instruction request by broadcasting the traffic mode to the plurality of processor clusters, the bus access module having a second output coupled to the inputs of the pluratlity of processor clusters and controlling access to instruction request submissions from the plurality of processor clusters by broadcasting a signal representing a low traffic mode responsive to an average number of instruction requests within a limit, wherein during a high traffic mode, the plurality of processor clusters submit instruction requests during empty slots, an instruction request responsive to a cache miss by a processor within the processor cluster; and

a memory access module, having an input coupled to receive signals representing a plurality of instruction requests, the memory access module forwarding non-conflicting instruction requests.

44. (Original) The instruction request arbiter of claim 43, wherein the bus access module resolves conflicts between instruction requests having different priority indications by selecting an instruction request with the highest priority.

45. (Original) The instruction request arbiter of claim 43, wherein the memory access module detects conflicting instruction requests where more than one instruction request is associated with the same group of cache sets of a means for storing.

46. (Original) The instruction request arbiter of claim 43, wherein the high priority is indicative of an instruction with a critical instruction request.

47. (Previosly presented) The instruction request arbiter of claim 43, wherein the plurality of processor clusters each comprise a plurality of processing units, and the bus accessing means controls submissions of instruction requests by each processing unit

48. (Original) The instruction request arbiter of claim 47, wherein the plurality of processing units comprise network processing units and the instruction requests comprise requests of packet processing instructions related to at least one from the group consisting of: packet routing, switching, bridging, and forwarding.

49. (New) A processing unit, comprising:

a plurality of processor clusters, each processor cluster having an output to send a signal representing instruction requests, an instruction request responsive to a cache miss by a processing unit within the processor clusters;

an instruction request arbiter, having an input coupled to the outputs of the plurality of processor clusters, the instruction request arbiter controlling access of the plurality of processor clusters to submit instruction requests by broadcasting the traffic mode to the plurality of processor clusters, the instruction request arbiter having a second output coupled to the inputs of the pluratlity of processor clusters and controlling access to instruction request submissions from the plurality of processor clusters by broadcasting a signal representing a low traffic mode responsive to an average number of instruction requests within a limit, wherein during a low traffic mode, the processor clusters submit instruction requests during empty slots, the instruction request arbiter also detecting conflicts between the instruction requests; and

an instruction memory, having an input coupled to a first output of the instruction request arbiter, the instruction memory sending a signal representing instruction data to the plurality of processor clusters responsive to receiving non-conflicting instruction requests from the instruction request arbiter.

50. (New) A processing unit, comprising:

a plurality of means for cluster processing, each means for cluster processing sending instruction requests, an instruction request responsive to a cache miss by a means for processing within the means for cluster processing;

means for arbitration, coupled to the plurality of means for cluster processing, the means for arbitration controlling access of the plurality of means for cluster processing to submit instruction requests by broadcasting the traffic mode to the plurality of means for cluster processing, the means for arbitration having a second output coupled to the inputs of the pluratlity of means for cluster processing and controlling access to instruction request submissions from the plurality of means for cluster processing by broadcasting a signal representing a high traffic mode responsive to an average number of instruction requests exceeding a limit, wherein during a high traffic mode, the plurality of means for cluster processing submit instruction requests during scheduled slots, the means for arbitration also detecting conflicts between the instruction requests; and

means for storing, coupled to the means for arbitration, the means for storing sending instruction data to the plurality of means for cluster processing responsive to receiving non-conflicting instruction requests from the means for arbitration.

51. (New) In a processing unit having an instruction memory hierarchy, a method for distributing instructions to an plurality of processing units organized in a plurality processor clusters, comprising:

determining a traffic mode for the plurality of processor clusters, an instruction request responsive to a cache miss by a processing unit within a processor cluster;

controlling submissions of instruction requests by each of the processing units by broadcasting the traffic mode to the plurality of processor clusters, including broadcasting a signal representing a low traffic mode responsive to an average number of instruction requests within a limit, wherein during a low traffic mode, the plurality of processor clusters submit instruction requests during empty slots;

receiving instruction requests;

detecting conflicts between the instruction requests; and

sending instruction data responsive to non-conflicting instruction requests.